

A Self-Adaptive LSE Wirelength Model for VLSI Global Placement

Zhiping Lai¹, Zhipeng Huang², Zhen Chen², and Jianli Chen²

¹College of Mathematics and Computer Science, Fuzhou University, Fuzhou 350108, China

²Center for Discrete Mathematics and Theoretical Computer Science, Fuzhou University, Fuzhou 350108, China
zpinglai@163.com; N155420004@fzu.edu.cn; chenchen@fzu.edu.cn; jlchen@fzu.edu.cn;

Abstract—Ignoring some cells overlaps, the common objective of very large scale integration (VLSI) global placement problem is to minimize the total half-perimeter wirelength (HPWL). Due to the non-differentiability of HPWL, the differentiable log-sum-exponential (LSE) wirelength model has been widely used to approximate HPWL. In order to overcome the disadvantage of fixed parameter value in LSE, a self-adaptive LSE wirelength model has been presented in this paper. In each iteration of cells diffusion, the proposed wirelength model is dynamically updated according to the overflow of the circuit, and then a self-adaptive wirelength based nonlinear solver is used to solve the placement problem. Compared with a state-of-the-art LSE based placer, the experimental results show that our wirelength model not only can improve the solution quality, but also can reduce the runtime.

I. INTRODUCTION

In VLSI physical design, the placement problem is to place circuit cells into a placement area such that no cell overlaps with the others and the total HPWL is optimized [1]. As the technology node enters the deep nanometer era with billion-transistor integration, the performance of a placement tool becomes dominant on the overall circuit performance of an EDA tool [6].

Up to now, there are three major types of methods used in placement [2]: annealing-based methods, partitioning-based methods, and analytical-based techniques. Recent studies have shown that analytical placers achieve the best placement quality with good scalability.

The placement problem is very hard to address directly due to the large problem size. In analytical approaches, a placement flow contains global placement stage and detailed placement stage. The global placement stage computes the best possibly locations of cells by ignoring some overlaps, and the detailed placement stage legalizes and further refines a global placement result. Global placement stage is considered as the most significant placement stage, which dominates the overall quality of the placement result.

Since the HPWL is non-differentiable, diverse differentiable wirelength approximation functions has appeared in analytical methods, such as the log-sum-exponential (LSE) wirelength model [7], the weighted-average (WA) wirelength model [4], and so on. Among them, the LSE wirelength model is widely adopted in many placers. In order to overcome the disadvantage of the fixed parameter value in original LSE wirelength model, a self-adaptive LSE wirelength model has been presented in this paper, in which the value of wirelength parameter is updated according to the overflow in each iteration of cells diffusion. And a self-adaptive LSE based nonlinear solver is used to solve the placement problem, which improves the solution quality and reduces the runtime.

The remainder of this paper is organized as follows. Section II introduces the VLSI global placement problem. Section III shows

the self-adaptive LSE wirelength model followed by the self-adaptive LSE based solver. Experimental results are reported in Section IV. Finally, Section V gives some conclusions.

II. PROBLEM STATEMENT

Given n cells and r nets, a circuit of the VLSI placement problem can be modelled as a hypergraph $G(V, E)$, where cells are denoted by vertices $V = \{v_1, v_2, v_3, \dots, v_n\}$, and nets are denoted by hyperedges $E = \{e_1, e_2, e_3, \dots, e_r\}$. Let (w_i, h_i) be the respective width and height of cell v_i , and (x_i, y_i) be the coordinate of the center of the cell, $i = 1, 2, \dots, n$. Let the placement region be $[0, W] \times [0, H]$. The VLSI placement problem computes the best position for each cell such that no cell overlaps with the others, and the total wirelength is minimized [2]:

$$\min W(v) \quad \text{s.t.} \quad \text{no overlaps among cells}, \quad (1)$$

where $W(v)$ is the total wirelength calculated by the half-perimeter wirelength (HPWL) [2],

$$W(v) = \sum_{e \in E} \left(\max_{v_i, v_j \in e} |x_i - x_j| + \max_{v_i, v_j \in e} |y_i - y_j| \right). \quad (2)$$

Since the HPWL is not differentiable, it is difficult to optimize it directly. To smooth the HPWL well, a popular log-sum-exponential (LSE) wirelength model is presented in [7], which is:

$$W(v) = \gamma \sum_{e \in E} \left(\log \sum_{v_k \in e} e^{\frac{x_k}{\gamma}} + \log \sum_{v_k \in e} e^{\frac{-x_k}{\gamma}} + \log \sum_{v_k \in e} e^{\frac{y_k}{\gamma}} + \log \sum_{v_k \in e} e^{\frac{-y_k}{\gamma}} \right). \quad (3)$$

If the value of γ is small enough, the LSE wirelength model can approximate the HPWL exactly.

According to the large scale of problem size, the constraints are also difficult to deal with directly. Most of methods smooth the constraints by dividing the placement region into uniform bins. By using the penalty method, the placement problem is transformed into an unconstrained minimization optimization problem [2].

III. OUR ALGORITHM

A. Self-Adaptive LSE Wirelength Model

In original LSE wirelength model, a fixed value is set for wirelength parameter γ , the choice of γ has great influence on global placement result. If the γ is large, the LSE wirelength model can not approximate HPWL exactly while cells can spread quickly; on the contrary, a smaller one spreads cells slowly and the data may overflow while the model approximates HPWL

precisely. Usually, the fixed parameter value is set according to a benchmark and its experience.

To overcome the disadvantages of original LSE wirelength model, we present a self-adaptive LSE wirelength model to solve the global placement problem. In the proposed model, the value of γ is dynamically updated according to the overflow of a circuit in the process of cells diffusion. By dividing the placement region into $m \times m$ uniform bins, the overflow is calculated by:

$$\varepsilon = \frac{\sum_b \max(\rho_b - \rho_t, 0) A_b}{\sum_{i \in V_m} A_i}, \quad (4)$$

where ρ_b denotes the density of bin b , ρ_t is the target density, A_b is the area of bin b , and A_i is the area of movable cell i .

In order to void data overflow, the lower bound of γ is set. Furthermore, an upper bound is set to prevent from over-destruction of wirelength. The complete wirelength parameter function is presented as

$$\gamma = \begin{cases} \gamma_{\min}, & \varepsilon \leq \varepsilon_{\min}, \\ \gamma_{\min} \left(\frac{\gamma_{\min}}{\gamma_{\max}} \right)^{\frac{\varepsilon_{\min} - \varepsilon}{\varepsilon_{\max} - \varepsilon_{\min}}}, & \varepsilon_{\min} < \varepsilon \leq \varepsilon_{\max}, \\ \gamma_{\max}, & \varepsilon > \varepsilon_{\max}. \end{cases} \quad (5)$$

Fig. 1(a) gives an illustration that our wirelength parameter γ is dynamically updated according to the overflow. In Fig. 1(a), the values of $\varepsilon_{\min}$ and $\varepsilon_{\max}$ are often set to 0.10 and 0.25, respectively; $\gamma_{\min}$ and $\gamma_{\max}$ are determined by the size of a circuit. The lower bound $\gamma_{\min}$ is set to be $\frac{\max(W, H)}{709.08956571}$ according to [3], where $\max(W, H)$ is the maximum of chip width / height. For example, the maximum of chip width / height in one of the experimental ICCAD 2012 benchmarks is 30933, we can get $\gamma_{\min} = \frac{30933}{709.0895657} \approx 44$.

Fig. 1(b) shows the differences of the HPWL variation between self-adaptive LSE wirelength model and original LSE wirelength model. The algorithm is based on our two level framework which described in III-B, and we can see that our proposed model can achieve better HPWL and less iterations. In our self-adaptive LSE wirelength model, at early stage, more cells are encouraged to be globally moved away from the high-density regions by relaxing the wirelength parameter. At later stage, when local movement dominates, the wirelength parameter is reduced to make the wirelength approach HPWL. When the global placement meets terminal conditions, the legalization and the detailed placement are called to deal with the rest of the problem.

If we only consider the wire force along the X direction, according to the LSE function (3), the value of wire force (gradient) of cell i can be calculated as:

$$\left| \frac{\partial W}{\partial x_i} \right| = \left| \sum_{e \in E} \left(\frac{e^{\frac{x_i}{\gamma}}}{\sum_{v_k \in e} e^{\frac{x_k}{\gamma}}} - \frac{e^{-\frac{x_i}{\gamma}}}{\sum_{v_k \in e} e^{-\frac{x_k}{\gamma}}} \right) \right|. \quad (6)$$

In Fig. 2, suppose that each pin is located on the center of respective cell, the cell2 is just connected with two cells in a net, the positions of cell1, cell2 and cell3 are $x_0 - 100$, x_0 , and $x_0 + 1000$, respectively. Considering the wire force on cell2 generated by this net, we can get the wire force on cell2 is $\left| \frac{1}{e^{-100/\gamma} + 1 + e^{1000/\gamma}} - \frac{1}{e^{100/\gamma} + 1 + e^{-1000/\gamma}} \right|$ according to the wire force function (6). In original LSE wirelength model, a fixed wirelength parameter ($\gamma = 50$) is set in the whole process of global placement, the wire force on cell2 generated by this net

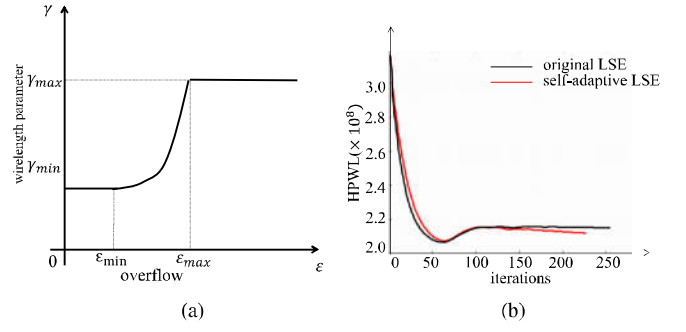


Fig. 1: (a) Our self-adaptive LSE wirelength function. (b) Compare our self-adaptive LSE model with original LSE model on benchmark superblue4. Compared with original LSE model, our self-adaptive LSE model can spread cells quickly in the previous iterations and approach HPWL well in the last iterations, which makes a better trade-off between runtime and solution quality.

is 0.119; In self-adaptive LSE wirelength model, a large value ($\gamma = 100$) is set at early stage, the wire force on cell2 is 0.497. Since 0.497 is larger than 0.119, the diffusion speed of cell2 in self-adaptive LSE wirelength model is faster, we can get the possible positions of cells more quickly.

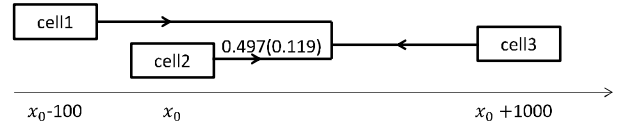


Fig. 2: The wire force on cell2 when $\gamma = 100$ ($\gamma = 50$).

At later stage, once the cells are scattered enough, the value of parameter decreases until reach the lower bound. According to function (3), when the γ is small enough, $\log \sum_{v_k \in e} e^{\frac{x_k}{\gamma}}$ and $\log \sum_{v_k \in e} e^{-\frac{x_k}{\gamma}}$ are approximately equal $\max_{v_i \in e} x_i$ and $-\min_{v_j \in e} x_j$, respectively. So we can get $W_x(v) = \max_{v_i, v_j \in e} |x_i - x_j|$ in X direction, and it is similar in Y direction, which means a good approximation of HPWL. In our model, since the diffusion speed of cells and the accuracy of approximation are considered together, we can not only improve the solution quality, but also reduce the runtime.

B. Self-Adaptive Wirelength Based Solver

Since the scale of the problem reaches millions of objects, some analytical placer (e.g., NTUplace4dr [5]) use the multilevel framework to reduce the size of the problem. In this paper, a two level framework is proposed in placement algorithm.

As shown in Algorithm 1, we propose our two level global placement algorithm. First, the clustering stage is in lines 1-3. For a given circuit information and the maximum cell number in clustering stage, we cluster cells to reduce the complexity of global placement problem. In line 4, we use the quadratic programming method to generate initial positions of the clusters without constraints. Lines 5-22 are the declustering stage. In line 10-11, we used the modified LSE wirelength model which proposed in Section III-A. In line 12, we use the penalty method to spread cells.

Algorithm 1 Algorithm Flow

Input:

A hypergraph $G(V, E)$: a circuit.

n_{max} : the maximum cell number in clustering stage.

Output:

optimal position x^*

```
1: while(CellNumber >  $n_{max}$ )
2:   call FirstChoiceClustering;
3: end while
4: initialize the cell position;
5: level = 1;
6: for current level = 1 to 2
7:   k=0;
8:   do
9:     compute density gradient;
10:    compute wirelength gradient  $\nabla W^k$ ;
11:    compute objective function;
12:    using optimization method to solve the problem;
13:    compute overflow  $\varepsilon$ ;
14:    update the penalty factor and wirelength parameters  $\gamma$ ;
15:    k=k+1;
16:  until  $\varepsilon < \varepsilon_{min}$  and not further improve.
17:  if current level = 2
18:    return  $x^k$ 
19:  else
20:    decluster and update cell positions.
21:  end if
22: end for
```

IV. EXPERIMENTAL RESULTS

In this section, we compare our proposed placer with a leading placer NTUplace4dr [5]. Thanks to the codes provided by the authors in [5], all experiments in this paper were performed on the same platform with an Intel Xeon 2.4GHz Linux workstation and 64GB memory. In our experiments, we used the ICCAD 2012 [8] for our comparative studies based on wirelength and runtime.

Table I shows the circuit statistics of the ICCAD 2012 benchmarks: the numbers of movable cells, the HPWL of the final placement and the runtime, which are denoted by “#Mov.”, “HPWL” and “CPU(s)”, respectively. The bottom row “Normalized” gives the normalized wirelength ratio and runtime ratio based on our placement method.

TABLE I: HPWL ($\times E8$) and runtime comparison based on the ICCAD 2012 benchmarks

Benchmarks	#Mov.	Ntuplace4dr		Our placer	
		HPWL	CPU(s)	HPWL	CPU(s)
superblue1	765k	3.16	3150	2.60	1224
superblue3	833k	3.38	2376	2.97	1257
superblue4	521k	2.35	1402	2.10	627
superblue5	677k	3.77	2750	3.40	1733
superblue7	1271k	4.75	4570	3.67	2266
superblue10	1046k	5.69	3075	5.24	1598
superblue16	680k	3.02	1285	2.59	882
superblue18	442k	1.50	2230	1.35	991
Normalized		1.155	1.970	1.000	1.000

Compared with NTUplace4dr, from Table I we can see that our placer achieves better HPWL results and shorter runtimes on all the eight benchmarks. On average, we achieve 15.5% smaller HPWL and $1.97\times$ faster than NTUplace4dr. Fig. 3 illustrates the cell distribution for initial positions and final positions of cells on the superblue4 benchmark. Standard cells and macros are denoted by blue and yellow rectangles, respectively.

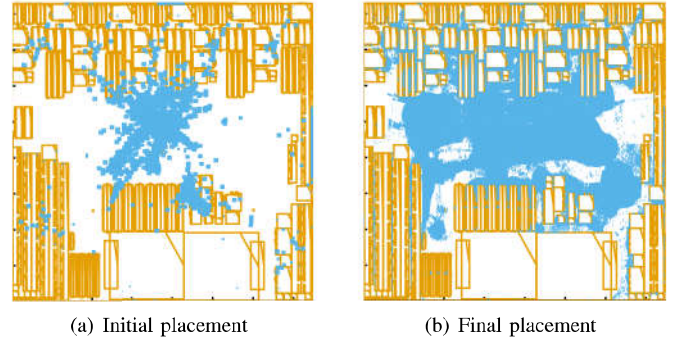


Fig. 3: Cells distribution for initial positions and final positions on the superblue4 benchmark. Standard cells and macros are denoted by blue and yellow rectangles, respectively.

V. CONCLUSIONS

In this paper, we propose a self-adaptive wirelength model to well solve the VLSI global placement problem. Based on the LSE wirelength, our model can adaptively adjust according to the density distribution. This model can achieve better result and shorter runtime. And we use a two level framework for our global placement algorithm to quickly obtain a better result. Compared with the leading placer NTUplace4dr, the experimental results show that our algorithm is efficient and effective.

VI. ACKNOWLEDGMENTS

The authors are thankful to Prof. Yao-Wen Chang and his EDA Lab group, National Taiwan University, for providing the source code of NTUplace2 and the binary of NTUplace4dr. This work was supported by the NSFC under Grants 11501115 and 61672005.

REFERENCES

- [1] C. J. Alpert and D. P. Mehta. *Handbook of Algorithms for Physical Design Automation*. 2008.
- [2] Y. W. Chang, Z. W. Jiang, and T. C. Chen. Essential issues in analytical placement algorithms. *IPSS Transactions on System LSI Design Methodology*, 2(4):145–166, 2009.
- [3] J. Chen and W. Zhu. A scaled lse wirelength model for vlsi global placement. In *International Conference on Fuzzy Systems and Knowledge Discovery*, pages 881–885, 2014.
- [4] M. K. Hsu, V. Balabanov, and Y. W. Chang. TSV-Aware analytical placement for 3-D IC designs based on a novel weighted-average wirelength model. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 32(4):497–509, 2013.
- [5] C. C. Huang, H. Y. Lee, B. Q. Lin, S. W. Yang, C. H. Chang, S. T. Chen, Y. W. Chang, T. C. Chen, and I. Bustany. NTUplace4dr: A detailed-routing-driven placer for mixed-size circuit designs with technology and region constraints. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, PP(99):1–1, 2017.
- [6] J. Lu, P. Chen, C. C. Chang, L. Sha, D. J. H. Huang, C. C. Teng, and C. K. Cheng. ePlace: electrostatics based placement using fast Fourier transform and Nesterov’s method. *ACM Transactions on Design Automation of Electronic Systems*, 20(2):17:1–17:34, 2015.
- [7] W. C. Naylor, R. Donnelly, and L. Sha. Non-linear optimization system and method for wire length and delay optimization for an automatic electric circuit placer. 2001.
- [8] N. Viswanathan, C. Alpert, C. Sze, Z. Li, and Y. Wei. ICCAD-2012 CAD contest in design hierarchy aware routability-driven placement and benchmark suite. In *Proceedings of IEEE/ACM International Conference on Computer-Aided Design*, pages 345–348, 2012.